18 JAN 2005

PCT/JP03/09153

日本国特許庁
JAPAN PATENT OFFICE

15.08.03

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。
This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: ２００２年　７月１９日

出願番号
Application Number: 特願２００２－２１１０２１

[ST. 10/C]: [ＪＰ２００２－２１１０２１]

出願人
Applicant(s): 独立行政法人産業技術総合研究所

２００３年　９月１９日

特許庁長官
Commissioner,
Japan Patent Office
今井康夫

出証番号　出証特２００３－３０７６９５５

【書類名】　特許願
【整理番号】　113MS0348
【提出日】　平成14年 7月19日
【あて先】　特許庁長官殿
【国際特許分類】　G06F 17/00
【発明者】
　【住所又は居所】　兵庫県尼崎市若王寺３丁目１１番４６号　独立行政法人産業技術総合研究所関西センター尼崎事業所内
　【氏名】　大崎　人士
【発明者】
　【住所又は居所】　兵庫県尼崎市若王寺３丁目１１番４６号　独立行政法人産業技術総合研究所関西センター尼崎事業所内
　【氏名】　高井　利憲
【特許出願人】
　【識別番号】　301021533
　【氏名又は名称】　独立行政法人産業技術総合研究所
　【代表者】　理事長　吉川　弘之
　【連絡先】　０７２７－５１－９６８１
【提出物件の目録】
　【物件名】　明細書　1
　【物件名】　図面　1
　【物件名】　要約書　1
【プルーフの要否】　要

【書類名】　明細書

【発明の名称】　リアクティブ・システムの安全性検証装置、方法、プログラム及びそのプログラムを記録した記録媒体

【特許請求の範囲】

【請求項１】　関数記号の集合、書換規則の集合、公理の集合、項の集合、及び検査対象となる項の集合によって表わされるリアクティブ・システムの安全性検証装置であって、

前記公理の集合が、交換則及び結合則のみを要素とする集合であり、

前記公理の集合の下で、前記項の集合を受理する第１の等式付ツリー・オートマトンを生成する翻訳部、

前記第１の等式付ツリー・オートマトンを初期データとして、前記書換規則の集合及び前記公理の集合の下で、前記項の集合から派生する項からなる集合と前記項の集合とを受理する第２の等式付ツリー・オートマトンを生成するシミュレーション部、及び

前記第２の等式付ツリー・オートマトンと、前記検査対象となる項の集合を受理する第３の等式付ツリー・オートマトンとを結合して第４の等式付ツリー・オートマトンを生成し、該第４の等式付ツリー・オートマトンが受理する集合が空集合か否かを判断する集合演算部を備えていることを特徴とするリアクティブ・システムの安全性検証装置。

【請求項２】　関数記号の集合、書換規則の集合、公理の集合、項の集合、及び検査対象となる項の集合によって表わされるリアクティブ・システムの安全性検証装置であって、

前記公理の集合が、交換則及び結合則のみを要素とする集合であり、

前記公理の集合の下で、前記項の集合を受理する第１の等式付ツリー・オートマトンを生成する翻訳部、

前記第１の等式付ツリー・オートマトンを初期データとして、前記書換規則の集合及び前記公理の集合の下で、前記項の集合から派生する項からなる集合と前記項の集合とを受理する第２の等式付ツリー・オートマトンを生成するシミュレーション部、及び

前記第２の等式付ツリー・オートマトンが、前記検査対象となる項を受理するか否かを判断する集合演算部を備えていることを特徴とするリアクティブ・システムの安全性検証装置。

【請求項３】　前記関数記号の集合は、暗号処理、復号処理、及び通信処理を表す関数を要素として含む集合であり、

前記書換規則の集合は、暗号処理された情報が復号処理されることによって平文に戻ることを表す規則を要素として含む集合であり、

前記検査対象となる項は、秘密情報であり、

前記項の集合は、秘密情報を交換する複数の主体の各々の知識の集合、及びこれら複数の主体間で交換される情報を傍受する主体の知識の集合であることを特徴とする請求項１又は２に記載のリアクティブ・システムの安全性検証装置。

【請求項４】　関数記号の集合、書換規則の集合、公理の集合、項の集合、及び検査対象となる項の集合によって表わされるリアクティブ・システムの安全性検証方法であって、

前記公理の集合が、交換則及び結合則のみを要素とする集合であり、

前記公理の集合の下で、前記項の集合を受理する第１の等式付ツリー・オートマトンを生成する第１のステップ、

前記第１の等式付ツリー・オートマトンを初期データとして、前記書換規則の集合及び前記公理の集合の下で、前記項の集合から派生する項からなる集合と前記項の集合とを受理する第２の等式付ツリー・オートマトンを生成する第２のステップ、及び

前記第２の等式付ツリー・オートマトンと、前記検査対象となる項の集合を受理する第３の等式付ツリー・オートマトンとを結合して第４の等式付ツリー・オートマトンを生成し、該第４の等式付ツリー・オートマトンが受理する集合が空集合か否かを判断する第３のステップを含むことを特徴とするリアクティブ・システムの安全性検証方法。

【請求項５】　関数記号の集合、書換規則の集合、公理の集合、項の集合、及び検査対象となる項の集合によって表わされるリアクティブ・システムの安全性検証方法であって、

前記公理の集合が、交換則及び結合則のみを要素とする集合であり、

前記公理の集合の下で、前記項の集合を受理する第１の等式付ツリー・オートマトンを生成する第１のステップ、

前記第１の等式付ツリー・オートマトンを初期データとして、前記書換規則の集合及び前記公理の集合の下で、前記項の集合から派生する項からなる集合と前記項の集合とを受理する第２の等式付ツリー・オートマトンを生成する第２のステップ、及び

前記第２の等式付ツリー・オートマトンが、前記検査対象となる項を受理するか否かを判断する第３のステップを含むことを特徴とするリアクティブ・システムの安全性検証方法。

【請求項６】　前記関数記号の集合は、暗号処理、復号処理、及び通信処理を表す関数を要素として含む集合であり、

前記書換規則の集合は、暗号処理された情報が復号処理されることによって平文に戻ることを表す規則を要素として含む集合であり、

前記検査対象となる項は、秘密情報であり、

前記項の集合は、秘密情報を交換する複数の主体の各々の知識の集合、及びこれら複数の主体間で交換される情報を傍受する主体の知識の集合であることを特徴とする請求項４又は５に記載のリアクティブ・システムの安全性検証方法。

【請求項７】　コンピュータに、

関数記号の集合、書換規則の集合、公理の集合、項の集合、及び検査対象となる項の集合によって表わされた手順の入力を受け付ける機能、

交換則及び結合則のみを要素とする前記公理の集合の下で、前記項の集合を受理する第１の等式付ツリー・オートマトンを生成する機能、

前記第１の等式付ツリー・オートマトンを初期データとして、前記書換規則の集合及び前記公理の集合の下で、前記項の集合から派生する項からなる集合と前記項の集合とを受理する第２の等式付ツリー・オートマトンを生成する機能、及び

前記第２の等式付ツリー・オートマトンと、前記検査対象となる項の集合を受理する第３の等式付ツリー・オートマトンとを結合して第４の等式付ツリー・オ

ートマトンを生成し、該第４の等式付ツリー・オートマトンが受理する集合が空集合か否かを判断する機能を実現させるためのリアクティブ・システムの安全性検証プログラム。

【請求項８】　コンピュータに、

関数記号の集合、書換規則の集合、公理の集合、項の集合、及び検査対象となる項の集合によって表わされた手順の入力を受け付ける機能、

交換則及び結合則のみを要素とする前記公理の集合の下で、前記項の集合を受理する第１の等式付ツリー・オートマトンを生成する機能、

前記第１の等式付ツリー・オートマトンを初期データとして、前記書換規則の集合及び前記公理の集合の下で、前記項の集合から派生する項からなる集合と前記項の集合とを受理する第２の等式付ツリー・オートマトンを生成する機能、及び

前記第２の等式付ツリー・オートマトンが、前記検査対象となる項を受理するか否かを判断する機能を実現させるためのリアクティブ・システムの安全性検証プログラム。

【請求項９】　前記関数記号の集合は、暗号処理、復号処理、及び通信処理を表す関数を要素として含む集合であり、

前記書換規則の集合は、暗号処理された情報が復号処理されることによって平文に戻ることを表す規則を要素として含む集合であり、

前記検査対象となる項は、秘密情報であり、

前記項の集合が、秘密情報を交換する複数の主体の各々の知識の集合、及びこれら複数の主体間で交換される情報を傍受する主体の知識の集合であることを特徴とする請求項７又は８に記載のリアクティブ・システムの安全性検証プログラム。

【請求項１０】　請求項７～９の何れかの項に記載のプログラムを記録したコンピュータ読取可能な記録媒体。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、ツリー・オートマトン理論に基づきリアクティブ・システムの安全性を検証する装置、方法、プログラム及びそのプログラムを記録した記録媒体に関する。

【０００２】

【従来の技術】

近年、コンピュータ技術、通信技術の進歩に伴い、様々な産業分野において、公衆ネットワーク及び専用ネットワークを介して大量の情報交換が行われており、秘密情報の交換が必要となる場合が多く発生している。特に、金融ビジネス、電子商取引などの分野においては、通信者間の認証、通信情報の機密保持などが高い精度で要求されており、暗号処理などの安全性を確保するための各種方法が開発され、利用されている。提供されるサービスの安全性は、暗号の堅牢性が保証されている場合には、主として使用される暗号通信手順の安全性（例えば、暗号化情報を復号化処理する場合に用いる「秘密鍵」が、意図する相手にのみ受け渡しされていることの確実性）に依存することとなる。従って、開発された暗号通信手順の安全性の検証が、秘密情報の秘匿性を保証する上で、非常に重要な技術となる。

【０００３】

本願において、「暗号通信手順」には、データを暗号処理、復号処理する手順、及び暗号化されたデータが通信回線を介して交換される手順が含まれるが、通信規格が定めるデータグラムのビット形式や通信経路の動的制御などの実装の方式は含まれない。

【０００４】

また、「安全性検証」とは、暗号通信手順に限らず、動作中に外界からの刺激を受けて、その刺激に対する応答を返す動作を繰り返すシステムであるリアクティブ・システムの動作手順を対象として、それが意図通りに記述されているか否かを確認することを指し、システムが如何なる場合にも意図しない状態、例えば危険な状態にならないとき、そのシステムが「安全」であるという。暗号情報を通信するシステムもリアクティブ・システムの一種であり、暗号通信手順をシステムの動作手順とみなすことができる。暗号通信手順の安全性検証においては、

実際の通信回線の電気的な信頼性及び品質は、検証の対象外であり、安全性とは交換される情報の秘密保持性を意味する。

【０００５】

従来、暗号通信手順の検証方法として、オートマトン理論に基づく正則ツリー・オートマトンと呼ばれる枠組みを用いた方法が知られている。初期に提案された検証方法は、楫勇一、藤原融、嵩忠雄による“Solving a Unification Problem under Constrained Substitutions Using Tree Automata(ツリー・オートマトンを用いた制約付代入における単一化問題の解法)”（Journal of Symbolic Computation 23(1), pp.79-117, 1997）に開示されている。

【０００６】

上記の方法を発展させた方法が、David Monniauxによる“Abstracting Cryptographic Protocols with Tree Automata(ツリー・オートマトンによる暗号通信プロトコルの抽出化法)”（Proceeding of 6th International Static Analysis Symposium, Venice(Italy), Lecture Notes in Computer Science 1694, pp.149-163, 1999）、及びThomas Genet, Francis Klayによる“Rewriting for Cryptographic Protocol Verification(暗号通信プロトコル検証のための書換系)”（Proceeding of 17th International Conference on Automated Deduction, Pittsburgh(PA), Lecture Notes in Computer Science 1831, pp.271-290, 2000）に開示されている。

【０００７】

オートマトンとは、実際の装置、システムなどを抽象的に表現した系であり、複数の状態を取ることができ、「入力」によって各状態間の遷移が起こる。取り得る状態は、必ずしも有限とは限らない。１つ又は一連の複数の入力ＩＮＰＵＴによって、オートマトンが、初期状態から所定の終了状態に至った場合、ＩＮＰＵＴがオートマトンによって受理されたという。一般に、オートマトンは（Σ、Ｑ、Ｑｆ、Δ）と記述される。ここで、Σは入力の集合、Ｑは取り得る状態の集合、Ｑｆは終了状態の集合、Δは遷移規則の集合である。

【０００８】

従って、ある集合の構成要素だけを受理し、その他を受理しないオートマトン

を論理式として記述することができれば、その集合に対する処理、即ちその集合の要素に対する処理を、その論理式を用いて等価的に行うことができる。このことは、処理対象の集合が無限の要素からなる場合に、特に有効である。

【０００９】

ツリー・オートマトンとは、ツリー構造を有するデータを受理するオートマトンを表す。また、正則ツリー・オートマトンとは、正則性の条件を満たすツリー・オートマトンを表す。

【００１０】

【発明が解決しようとする課題】

検証の対象のひとつである暗号通信手順を形式言語で表現する場合には、正則性の条件を満すことが必要であった。このために、従来までの形式言語理論によるアプローチでは、正則性の条件を満たさない暗号通信手順を、自動的に検証することはできなかった。

【００１１】

上記した３つの論文に提案されているいずれの検証方法も、正則性の条件を満たさない暗号通信手順については、秘密保持性を近似的に検証することは可能であっても、厳密な検証を行うことはできないという問題がある。

【００１２】

このことは、暗号通信手順に限らず、一般的なリアクティブ・システムの動作手順に関しても生じる問題である。

【００１３】

本発明は、正則性の条件を満たすか否かによらず、等式付ツリー・オートマトン理論の範疇にあるリアクティブ・システムの動作手順に関して、近似ではなく、厳密に安全性を検証することができるリアクティブ・システムの安全性検証装置、検証方法、検証プログラム及びそのプログラムを記録したコンピュータ読取可能な記録媒体を提供することを目的とする。

【００１４】

【課題を解決するための手段】

本発明の目的は、以下の手段によって達成される。

【００１５】

即ち、本発明の第１の態様によれば、関数記号の集合、書換規則の集合、公理の集合、項の集合、及び検査対象となる項の集合によって表わされるリアクティブ・システムの安全性検証装置であって、前記公理の集合が、交換則及び結合則のみを要素とする集合であり、前記公理の集合の下で、前記項の集合を受理する第１の等式付ツリー・オートマトンを生成する翻訳部、前記第１の等式付ツリー・オートマトンを初期データとして、前記書換規則の集合及び前記公理の集合の下で、前記項の集合から派生する項からなる集合と前記項の集合とを受理する第２の等式付ツリー・オートマトンを生成するシミュレーション部、及び前記第２の等式付ツリー・オートマトンと、前記検査対象となる項の集合を受理する第３の等式付ツリー・オートマトンとを結合して第４の等式付ツリー・オートマトンを生成し、該第４の等式付ツリー・オートマトンが受理する集合が空集合か否かを判断する集合演算部を備えていることを特徴とするリアクティブ・システムの安全性検証装置を提供することができる。

【００１６】

本発明の第２の態様によれば、関数記号の集合、書換規則の集合、公理の集合、項の集合、及び検査対象となる項の集合によって表わされるリアクティブ・システムの安全性検証装置であって、前記公理の集合が、交換則及び結合則のみを要素とする集合であり、前記公理の集合の下で、前記項の集合を受理する第１の等式付ツリー・オートマトンを生成する翻訳部、前記第１の等式付ツリー・オートマトンを初期データとして、前記書換規則の集合及び前記公理の集合の下で、前記項の集合から派生する項からなる集合と前記項の集合とを受理する第２の等式付ツリー・オートマトンを生成するシミュレーション部、及び前記第２の等式付ツリー・オートマトンが、前記検査対象となる項を受理するか否かを判断する集合演算部を備えていることを特徴とするリアクティブ・システムの安全性検証装置を提供することができる。

【００１７】

本発明の第３の態様によれば、関数記号の集合、書換規則の集合、公理の集合、項の集合、及び検査対象となる項の集合によって表わされるリアクティブ・シ

ステムの安全性検証方法であって、前記公理の集合が、交換則及び結合則のみを要素とする集合であり、前記公理の集合の下で、前記項の集合を受理する第１の等式付ツリー・オートマトンを生成する第１のステップ、前記第１の等式付ツリー・オートマトンを初期データとして、前記書換規則の集合及び前記公理の集合の下で、前記項の集合から派生する項からなる集合と前記項の集合とを受理する第２の等式付ツリー・オートマトンを生成する第２のステップ、及び前記第２の等式付ツリー・オートマトンと、前記検査対象となる項の集合を受理する第３の等式付ツリー・オートマトンとを結合して第４の等式付ツリー・オートマトンを生成し、該第４の等式付ツリー・オートマトンが受理する集合が空集合か否かを判断する第３のステップを含むことを特徴とするリアクティブ・システムの安全性検証方法を提供することができる。

【００１８】

本発明の第４の態様によれば、関数記号の集合、書換規則の集合、公理の集合、項の集合、及び検査対象となる項の集合によって表わされるリアクティブ・システムの安全性検証方法であって、前記公理の集合が、交換則及び結合則のみを要素とする集合であり、前記公理の集合の下で、前記項の集合を受理する第１の等式付ツリー・オートマトンを生成する第１のステップ、前記第１の等式付ツリー・オートマトンを初期データとして、前記書換規則の集合及び前記公理の集合の下で、前記項の集合から派生する項からなる集合と前記項の集合とを受理する第２の等式付ツリー・オートマトンを生成する第２のステップ、及び前記第２の等式付ツリー・オートマトンが、前記検査対象となる項を受理するか否かを判断する第３のステップを含むことを特徴とするリアクティブ・システムの安全性検証方法を提供することができる。

【００１９】

本発明の第５の態様によれば、コンピュータに、関数記号の集合、書換規則の集合、公理の集合、項の集合、及び検査対象となる項の集合によって表わされた手順の入力を受け付ける機能、交換則及び結合則のみを要素とする前記公理の集合の下で、前記項の集合を受理する第１の等式付ツリー・オートマトンを生成する機能、前記第１の等式付ツリー・オートマトンを初期データとして、前記書換

規則の集合及び前記公理の集合の下で、前記項の集合から派生する項からなる集合と前記項の集合とを受理する第２の等式付ツリー・オートマトンを生成する機能、及び前記第２の等式付ツリー・オートマトンと、前記検査対象となる項の集合を受理する第３の等式付ツリー・オートマトンとを結合して第４の等式付ツリー・オートマトンを生成し、該第４の等式付ツリー・オートマトンが受理する集合が空集合か否かを判断する機能を実現させるためのリアクティブ・システムの安全性検証プログラムを提供することができる。

【００２０】

本発明の第６の態様によれば、コンピュータに、関数記号の集合、書換規則の集合、公理の集合、項の集合、及び検査対象となる項の集合によって表わされた手順の入力を受け付ける機能、交換則及び結合則のみを要素とする前記公理の集合の下で、前記項の集合を受理する第１の等式付ツリー・オートマトンを生成する機能、前記第１の等式付ツリー・オートマトンを初期データとして、前記書換規則の集合及び前記公理の集合の下で、前記項の集合から派生する項からなる集合と前記項の集合とを受理する第２の等式付ツリー・オートマトンを生成する機能、及び前記第２の等式付ツリー・オートマトンが、前記検査対象となる項を受理するか否かを判断する機能を実現させるためのリアクティブ・システムの安全性検証プログラムを提供することができる。

【００２１】

前記第１～第６の態様において、前記関数記号の集合が、暗号処理、復号処理、及び通信処理を表す関数を要素として含む集合であり、前記書換規則の集合が、暗号処理された情報が復号処理されることによって平文に戻ることを表す規則を要素として含む集合であり、前記検査対象となる項が、秘密情報であり、前記項の集合が、秘密情報を交換する複数の主体の各々の知識の集合、及びこれら複数の主体間で交換される情報を傍受する主体の知識の集合であることができる。

【００２２】

本発明の第７の態様によれば、上記した何れかのプログラムを記録したコンピュータ読取可能な記録媒体を提供することができる。

【００２３】

【発明の実施の形態】

以下、本発明に係る実施の形態に関して、添付図を参照して説明する。図１は、本発明の実施の形態に係るリアクティブ・システムの安全性検証装置の構成を示すブロック図である。本実施の形態に係るリアクティブ・システムの安全性検証装置は、各構成部を制御する中央演算処理部（以下、ＣＰＵと記す）１、データを一時的に記録する一時記憶部（以下、メモリと記す）２、データを持続的に記録する記録部３、各構成部の間でデータを交換するための内部バス４、データの入力を受け付ける入力部５、処理結果などを表示する表示部６、インタフェース部７、翻訳部８、シミュレーション部９、及び集合演算部１０を備えている。

【００２４】

ＣＰＵ１は、各構成部に対する制御を行い、翻訳部８、シミュレーション部９、及び集合演算部１０は、ＣＰＵ１からの命令を受けて、後述するように安全性検証における中心的処理を行う。検証の対象である動作手順に関する情報は、入力部５から入力され、インタフェース部７によって所定のデータ形式に変換され、記録部３に記録される。後述するデータ処理において、記録部３から必要なデータがメモリ２に読み出され、メモリ２上で処理が実行される。処理結果は記録部３に記録され、必要に応じて表示部６に表示される。

【００２５】

図２は、暗号通信手順に関して、本実施の形態に係るリアクティブ・システムの安全性検証装置の処理を示すフローチャートである。図１及び２に基づき、検証装置の処理内容を説明する前に、検証の対象のひとつである暗号通信手順の記述方法に関して説明する。

【００２６】

有限の要素からなる集合を対象とした処理は、直接各要素を記述し、これに対する処理を行うことが可能であるが、本発明が検証の対象とする暗号通信手順は、一般に有限の要素からなる集合として記述することはできない。従って、検証の対象を、取り得る状態の集合と、状態間の遷移を引き起こす入力とによって記述された「系」、即ちオートマトンを用いて表す。

【００２７】

具体的には、検証の対象となる暗号通信手順を次の式１に示す５個の記号の組で表現する。このことは、式１の記号表現によって表現することが可能な「通信手順」のみが検証の対象となり得ることを意味するものでもある。以下において、記号｛｝は集合を表わし、例えば｛Ｌｉ｝はＬ１、Ｌ２、Ｌ３、などを構成要素とする集合を表すこととする。

Ｐ＝（Ｆ，｛Ｒｉ｝，Ｕ，｛Ｋｉ｝，Ｓ［ｊ→ｋ］）・・・・（式１）

ここで、Ｐは暗号通信手順を表わし、ｉは、ｍを自然数として、１≦ｉ≦ｍを満たし、ｊ、ｋはｍ以下の任意の正の整数とする。

【００２８】

Ｆ、Ｒｉ、Ｕ、Ｋｉ、Ｓ［ｊ→ｋ］は、全て集合を表す。Ｆは暗号処理、復号処理、通信処理（鍵交換、メッセージ交換など）、通信データ（メッセージ、パスワードなど）などを表す「関数記号」の集合、Ｒｉ（ｉ＝１～ｍ）は「書換規則」の集合、Ｕは「公理」の集合、Ｋｉ（ｉ＝１～ｍ）は「項」の集合（例えば、情報の交換に関係する主体がＡ、Ｂ、Ｃの３者であれば、各々に関してＫ１、Ｋ２、Ｋ３が設定される）を表わしている。ここで、「項」とは、集合の構成要素を意味し、単一の記号または複数の記号の組合せで表現される。Ｓ［ｊ→ｋ］は、Ｋｊを所有する主体がＫｋを所有する主体に対して秘密にしたいメッセージの集合（以下、検査対象となる項の集合と記す）を表している。

【００２９】

公理の集合Ｕは、交換則、結合則を要素とする集合であり、項ｘ、ｙに対する演算子を「＋」とすると、

交換則：ｘ＋ｙ＝ｙ＋ｘ

結合則：（ｘ＋ｙ）＋ｚ＝ｘ＋（ｙ＋ｚ）

と表すことができる。ここで「＋」は加算を表す記号ではない。

【００３０】

集合Ｋｉは、暗号情報を交換する複数の主体及び通信の傍受者の各々が獲得できる知識の集合を表わしている（以下、ナレッジと記す）。例えば、主体Ａ、Ｂが暗号情報を相互に交換する場合、主体Ａ、Ｂのナレッジを各々Ｋ１、Ｋ２とし、傍受者Ｃのナレッジを Ｋ３と表す。ナレッジは初期状態では有限個数の項の集

合であるが、一般に時間経過によって拡張される。例えば傍受者Ｃのナレッジ Ｋ３は、傍受者Ｃが主体Ａ、Ｂの間で繰り返し交換される暗号情報を傍受することに伴って、増加する。

【００３１】

書換規則｛Ｒｉ｝は、項に対して関数記号を適用した記述が、どのような異なる記述に書き換えられるかを指定した規則である。例えば、暗号処理をＥ、復号処理をＤ、鍵をｘ、平文メッセージをｙとすると、「暗号化メッセージをその暗号鍵で復号すると元の平文メッセージが得られる」という性質は、書換規則「Ｄ（ｘ，Ｅ（ｘ，ｙ））→ｙ」によって表わされる。ここで矢印は、矢印の左側の項を右側の項で書き換えることができることを表わしている。

【００３２】

以下において、図２に基づき、暗号通信手順に関して、本実施の形態に係るリアクティブ・システムの安全性検証装置が行う処理を説明する。

【００３３】

まず、ステップ２０において、ＣＰＵ１は、入力部５を介して、検証対象となる手順に関する情報、即ち上記した記号による表記（Ｆ，｛Ｒｉ｝，Ｕ，｛Ｋｉ｝，Ｓ［ｊ→ｋ］）の入力を受け付け、取得した入力データを記録部３に記録する。このとき、検査対象となる項の集合Ｓ［ｊ→ｋ］が無限集合の場合には、その集合を受理するオートマトンの記述が入力される。取得したデータは、論理記号を使用した論理式であり、例えばテキストデータ形式で入力され、以降の処理においてはテキスト形式のまま処理される。

【００３４】

ステップ２１において、ＣＰＵ１は、ステップ２０で記録部３に記録したデータを記録部３からメモリ２上の所定領域に読み出し、翻訳開始の命令コード及び必要なデータのメモリアドレス情報を翻訳部８に伝送する。

【００３５】

ステップ２２において、命令コードを受信した翻訳部８は、メモリアドレス情報に基づいてメモリ２からナレッジ｛Ｋｉ｝及び公理Ｕを取得し、各々のナレッジＫｉ及び公理Ｕを使用して等式付ツリー・オートマトンＡｉを生成する。生成

された等式付ツリー・オートマトンＡｉは、ＣＰＵ１を介してメモリ２の所定領域に記録される。

【００３６】

ナレッジＫｉの項を受理するオートマトンＡｉの生成方法は、楫勇一、藤原融、嵩忠雄による“Solving a Unification Problem under Constrained Substitution using Tree Automata(ツリー・オートマトンを用いた制約付代入における単一化問題の解法)”（Journal of Symbolic Computation 23(1), pp.79-117, 1997）などによって公知であり、この分野の通常の知識を有する者にとって容易であるので、ここでは記載を省略する。

【００３７】

翻訳部８が出力する各々のＡｉは、公理Ｕの下で、対応する集合であるナレッジＫｉを受理する等式付ツリー・オートマトンである。ここで、上記したように、オートマトンとは、取り得る状態と入力による状態間の遷移とによって、システムを表現した系であり、ツリー・オートマトンとは、ツリー構造を有するデータ（項）を受理するオートマトンである。

【００３８】

また、「等式付」とは、ツリー構造を有するデータの間に、等価性の概念を表現する公理が成立することを表している。例えば、項「１＋２」と項「２＋１」とを等価とみなすためには、公理「ｘ＋ｙ＝ｙ＋ｘ」が必要となる。同様に、項「（１＋２）＋３」と項「３＋（２＋１）」とを等価とみなすためには、ｘ＋ｙ＝ｙ＋ｘを仮定すれば十分である。公理「（ｘ＋ｙ）＋ｚ＝ｘ＋（ｙ＋ｚ）」が必要となるのは、例えば項「（１＋２）＋３」と項「１＋（２＋３）」とを等価とみなす場合である。ここで、「＋」は演算子であり、加算を意味するものではない。即ち、「等式付ツリー・オートマトン」とは、ツリー構造を有するデータの間に等価性の公理が成り立つことを前提として、ツリー構造のデータの受理／不受理を決定するオートマトンである。

【００３９】

また、「集合を受理する」とは、その集合の要素のみを受理する、即ち、その集合の要素（項）を全て受理し、それ以外を受理しないことを意味する。

【００４０】

以上のことから、各々のオートマトンＡｉは、各項がツリー構造を有し、項の間に等価性の公理Ｕが成り立つナレッジＫｉを受理する等式付ツリー・オートマトンとして記述される。

【００４１】

以上のように、ステップ２２において、翻訳部８は、各々の主体の初期のナレッジＫｉを等式付ツリー・オートマトンＡｉに変換、即ち、集合Ｋｉを受理する等式付ツリー・オートマトンＡｉとして記述する。これによって、集合Ｋｉに関する処理を、その集合と等価な等式付ツリー・オートマトンＡｉに関する処理として扱うことができる。

【００４２】

ステップ２３において、ＣＰＵ１は、シミュレーション開始の命令コードを、翻訳部８によって生成された等式付ツリー・オートマトン｛Ａｉ｝、ナレッジ｛Ｋｉ｝及び公理Ｕが記録されたメモリ２上のアドレス情報と共に、シミュレーション部９に伝送する。

【００４３】

ステップ２４において、ＣＰＵ１からの開始命令を受信したシミュレーション部９は、受信したメモリアドレス情報に基づいてメモリ２から等式付ツリー・オートマトン｛Ａｉ｝、ナレッジ｛Ｋｉ｝、公理Ｕを取得し、所定の処理を実行する。

【００４４】

即ち、シミュレーション部９は、メモリ２から読み出した等式付ツリー・オートマトン｛Ａｉ｝を初期データとして、繰り返し処理によって、拡張されるナレッジを受理する等式付ツリー・オートマトンを生成する。シミュレーション部９は、処理の経過途中において所定の収束条件が満たされたと判断した場合、例えば繰り返し処理の結果、等式付ツリー・オートマトンが変化しなくなったと判断した場合、その時点の等式付ツリー・オートマトン｛Ａｉ＊｝を出力し、収束したことを知らせる収束コードと共にＣＰＵ１に伝送する。ＣＰＵ１は、シミュレーション部９から受信したこれらの算出データを、メモリ２の所定領域に記録す

る。

【００４５】

図３は、上記した等式付ツリー・オートマトン｛Ａｉ＊｝の生成において、繰り返し処理で呼び出されるサブルーチンのアルゴリズムを示した図である。このアルゴリズムは、入出力引数の指定、初期設定、第１処理、第２処理から構成されている。入力引数は、書換規則の１つの要素（ｌ→ｒ）、及び等式付ツリー・オートマトン（Ａ/ＡＣ）であり、出力引数は計算結果の等式付ツリー・オートマトン（$B_{l\to r}$/ＡＣ）である。

【００４６】

初期設定では、以降の処理に対する初期値として、Ａ0に入力引数の値であるツリー・オートマトンＡをセットし、集合Ｓ、Ｔをセットする。集合Ｓ、Ｔは、入力引数の値である書換規則ｌ→ｒの左辺（ｌに相当する）、右辺（ｒに相当する）の項を、ツリー構造として記述した場合の位置情報を要素とする。

【００４７】

第１処理では、集合Ｓの要素に基づいて所定の順序で遷移規則を追加、変更する。第２処理では、集合Ｔの要素に基づいて所定の順序で処理して遷移規則を追加、変更する。第1処理及び第２処理は、同一の原理に基づいて計算を行うので、第1処理の場合についてのみ説明する。

【００４８】

第1処理において、まず、所定条件を満たす要素ｐを集合Ｓから選択する。この条件は、要素ｐがツリー構造の末端に位置する条件である。

【００４９】

次に、着目した要素ｐの関数記号をｆ、その引数部分の項を$t_1, \ldots, t_n$として、要素ｐに相当する部分の項がｆ（$t_1, \ldots, t_n$）と表せるとき、次の書換規則に従って、Ｌ（A_i/ＡＣ）に含まれるすべての項に対して書換を行うことによって、項の集合Ｌ（A_{i+1}/ＡＣ）を得る。

書換規則：ｆ（$c^{p.1}_{t1}, \ldots, c^{p.n}_{tn}$）→ $c^{p}_{l|p}$

項の集合Ｌ（A_{i+1}/ＡＣ）は、ｆが公理に用いられている関数記号である場合には、Rohit Parikhの手法をツリー構造のデータ用に拡張した方法で計算すること

ができる。ｆが公理に用いられている関数記号ではない場合には、上記した槇勇一らの論文に紹介されている方法で、A_iを基にA_{i+1}を計算することができる。以降繰り返し条件を満たすまで計算すれば、書換規則の左辺（ｌに相当する）に合致する項の前処理が終了し、次の処理に移る。

【００５０】

第２処理に関しても、所定条件を満たす要素ｑを集合Ｔから選択し、同様に処理することによって、第２処理のループ計算が繰り返し条件を満たして、等式付ツリー・オートマトンB_j/ACが得られるので、これを$B_{l\rightarrow r}$/ACとして第２処理を終了する。以上で$B_{l\rightarrow r}$/ACが得られ、これが出力引数の値となる。

【００５１】

以上の様に、ここでの処理には、Rohit Parikhによる文字列空間からベクトル空間への写像を用いた手法を、ツリー構造のデータを扱えるように開発した方法を用いている。Parikhの提案した方法は、"On Context-Free Languages"（Journal of the ACM 13(4), pp.570-581, 1966）に開示されているので、ここでは記載を省略する。

【００５２】

書換規則の複数の要素の各々に対して、等式付ツリー・オートマトンＡｉを入力引数A/ACとして、上記した処理を１回行い、それらの結果を集めることによって、１回の処理に相当して拡張されたナレッジを受理する等式付ツリー・オートマトンＡｉ＊(1)が得られる。得られた等式付ツリー・オートマトンＡｉ＊（１）を入力引数A/ACとして、再度同様に処理することによって、２回の処理に相当して拡張されたナレッジを受理する等式付ツリー・オートマトンＡｉ＊（２）が得られる。この処理を繰り返すことによって、次々と拡張されたナレッジを受理する等式付ツリー・オートマトンＡｉ＊（ｎ）が得られる。

【００５３】

シミュレータ部は、上記したように、等式付ツリー・オートマトンＡｉ＊（ｎ）が、１回前の等式付ツリー・オートマトンＡｉ＊（ｎ－１）から変化したか否かによって、収束を判断する。収束したと判断した場合、Ａｉ＊（ｎ）を、公理Ｕ及び書換規則Ｒｉの条件の下で、初期のナレッジＫｉから派生する全ての項か

らなる集合と初期のナレッジＫｉとを受理する等式付ツリー・オートマトンＡｉ＊とする。

【００５４】

主体が暗号情報を交換することによって、各々の主体及び傍受者のナレッジ｜Ｋｉ｜は徐々に増大して行くが、シミュレーション部９はステップ２４において、所定の条件（公理Ｕ及び書換規則｜Ｒｉ｜）の下で、この到達可能な最大のナレッジ｜Ｋｉ｜を等式付ツリー・オートマトン｜Ａｉ＊｜として記述する。これは、無限集合の境界を確定することに相当し、等式付ツリー・オートマトンとして記述することによって可能となる。

【００５５】

シミュレーション部９は、所定の時間若しくは所定回数の繰り返し処理を経過しても収束と判断できなかった場合、収束しなかったことを知らせる非収束コードをＣＰＵ１に伝送し、処理を終了する。これは、その暗号通信手順の検証ができないことを意味する。

【００５６】

ステップ２５において、ＣＰＵ１は、シミュレーション部９から受信したコードが、収束コードか否かを判断し、収束コードと判断した場合にステップ２６に移行し、非収束コードと判断した場合にはステップ３１に移行する。

【００５７】

ステップ２６において、ＣＰＵ１は、検査対象となる項の集合Ｓ［ｊ→ｋ］が有限集合か否かを判断し、有限集合と判断した場合、ステップ２７に移行し、有限集合でないと判断した場合、ステップ２８に移行する。ステップ２０において説明したように、検査対象となる項の集合Ｓ［ｊ→ｋ］が有限集合でない場合には、検査対象となる項の集合Ｓ［ｊ→ｋ］を受理する等式付ツリー・オートマトンの記述が入力されている。

【００５８】

秘密情報が有限である場合、ステップ２７において、演算部は、傍受者Ｃの最大ナレッジを表す等式付ツリー・オートマトンＡ３＊、及び検査対象となる項の集合Ｓ［ｊ→ｋ］、例えばＳ［１→３］をメモリ２から読み出し、各々の要素が

等式付ツリー・オートマトンＡ３＊に受理されるか否かを判断し、その結果に応じたデータをメモリ２の所定領域に記録する。このとき、検査対象となる項の集合Ｓ［１→３］のいずれの要素も等式付ツリー・オートマトンＡ３＊によって受理されなかった場合、記録されるデータは“０”であり、要素の中の少なくとも１つが等式付ツリー・オートマトンＡ３＊によって受理された場合、記録されるデータは“１”である。

【００５９】

検査対象となる項の集合Ｓ［ｊ→ｋ］が有限集合でない場合、ステップ２８において、ＣＰＵ１は集合演算部１０に開始命令及び必要なメモリアドレス情報を送信する。

【００６０】

ステップ２９において、集合演算部１０は、ＣＰＵ１から受信したメモリアドレス情報に基づいてメモリ２から、傍受者Ｃの可能な最大ナレッジを表す等式付ツリー・オートマトンＡ３＊、及び検査対象となる項の集合を受理する等式付ツリー・オートマトンＳ［ｊ→ｋ］、例えば主体Ａから傍受者Ｃに対する検査対象となる項の集合を受理する等式付ツリー・オートマトンＳ［１→３］を取得し、Ａ３＊とＳ［１→３］とを合成して等式付ツリー・オートマトンＷを生成する。この合成は、各々のオートマトンによって受理される２つの集合（Ａ３＊とＳ［１→３］）の積（Ａ３＊∩Ｓ［１→３］）を求めることに相当する。即ち、２つの集合の共通部分（Ａ３＊∩Ｓ［１→３］）は、等式付ツリー・オートマトンＷに受理される。

【００６１】

図４は、２つのツリー・オートマトンを合成する方法の一例を説明する図である。結合則及び交換則が成り立つ集合を受理する２つのオートマトン（A/AC、B/AC）を結合したオートマトンの遷移規則は、図４に示した４種類のＲｘ～Ｒｇを合わせたものとなる。詳細は、本願発明者による公知文献“Beyond Regularity：Equational Tree Automata for Associative and Commutative Theories(正則性を超えて：結合・交換理論に関する等式付ツリー・オートマトン)”（Proceedings of 15th International Conference of the European Association fo

r Computer Science Logic, Paris (France), Lecture Notes in Computer Science 2142, pp. 539-553, 2001.）において開示されているので、ここでは記載を省略する。

【００６２】

ステップ３０において、集合演算部１０は、ステップ２９で得られた等式付ツリー・オートマトンWが受理する集合Ａ３＊∩Ｓ［1→3］が空（くう：構成要素が存在しないこと）であるか否かを判断する。集合演算部１０は、空集合と判断すれば空コードをＣＰＵ１に伝送し、空集合でないと判断すれば非空コードをＣＰＵ１に伝送する。

【００６３】

空の判定は、項全体を受理する等式付ツリー・オートマトンB/ACの終了状態ｑから、判定対象である等式付ツリー・オートマトンA/ACの終了状態ｐに到達可能か否かを判断することによって行われる。具体的には、ツリー・オートマトンＢの遷移規則の左辺と右辺を入れ替え、これによって得られるツリー・オートマトンをB^{-1}とする。（Ａ∪B^{-1}）/ACを基底ＡＣ書換系とみなしたとき、状態ｑから状態ｐへの到達可能性は、等式付ツリー・オートマトンA/ACが何らかの要素を受理することと等価になる。

【００６４】

終了状態ｑから終了状態ｐに到達する経路を計算する具体的方法は、Richard MayerとMichael Rusinowitchによる“Reachability is Decidable for Ground AC Rewrite Systems（基底ＡＣ書換システムの到達可能性）”（Proceedings of 3rd International Workshop on Verification of Infinite State Systems, Aalborg (Denmark), 1998）に開示されているので、ここでは記載を省略する。

【００６５】

ステップ３１において、ＣＰＵ１は、表示部６に、以上のステップでの処理結果に応じた表示をする。即ち、ＣＰＵ１は、非収束コード（ステップ２５）を受信した場合、対象の暗号通信手順の検証ができないことを表示する。ＣＰＵ１は、ステップＳ２７での処理の結果としてメモリ２に記録されたデータが“０”（非受理を表す）の場合、検証対象の暗号通信手順が、その受理されなかった秘密

情報を通信する限りにおいて安全であることを表示し、“１”（受理を表す）の場合、検証対象の暗号通信手順が安全ではないことを表示する。ＣＰＵ１は、ステップ３０での処理の結果、空コードを受信した場合、検証対象の暗号通信手順が安全であることを表示し、非空コードを受信した場合、検証対象の暗号通信手順は安全ではないことを表示する。以上の処理の後、ＣＰＵ１は、必要に応じて、メモリ２上に一時記録されたデータを記録部３に記録し、暗号通信手順の安全性検証を終了する。

【００６６】

以下において、具体的な暗号通信手順への本発明の適用に関して説明する。

（実施例１）

本発明をDiffie-Hellman型暗号通信手順に適用する場合について説明する。Diffie-Hellman型暗号通信手順とは、主体Ａ、Ｂの間で暗号通信する場合に、暗号・復号用の鍵として秘密鍵と公開鍵の２種類の鍵を所定の規則に従って生成し、公開鍵を主体Ａ、Ｂの間で、例えば通信ネットワークを介して交換する暗号通信手順である。

【００６７】

Diffie-Hellman型暗号通信手順において、主体Ａは、任意に選択した大きい正の整数ｘを基に、次の式２によって公開鍵Ｘを生成し、Ｘを主体Ｂに通信する。主体Ｂは、任意に選択した大きい正の整数ｙを基に、次の式３によって公開鍵Ｙを生成し、Ｙを主体Ａに通信する。

$X = g^x \bmod n$　・・・・（式２）

$Y = g^y \bmod n$　・・・・（式３）

ここで、ｇ、ｎは任意の大きい素数であり、「ｍｏｄ」は剰余を表す。例えば、ａ ｍｏｄ ｂ は、ａをｂで除した場合の剰余を表わしている。

【００６８】

主体Ａは、主体Ｂから取得したＹ及び自己が選択したｘを使用して次の式４によってｋを計算することができ、また、主体Ｂは、主体Ａから取得したＸ及び自己が選択したｙを使用して次の式５によってｋ’を計算することができる。

$k = Y^x \bmod n$　・・・・（式４）

$k' = X^y \bmod n$　・・・・（式５）

計算結果のｋ及びｋ’は、何れも$g^{xy} \bmod n$に等しいことから、主体Ａ、Ｂは共通の数値を得ることができ、この値は、ｘまたはｙが分からなければ、主体Ａ、Ｂ間で交換されるｇ、ｎ、Ｘ、Ｙから取得することが非常に困難である。従って、主体Ａ、Ｂは、ｋ（＝ｋ’）を、暗号・復号用の鍵として使用することによって、安全に秘密データの交換が可能となる。

【００６９】

Diffie-Hellman型暗号通信手順は、上記した手順以外にも様々なバリエーションがあるが、それらは全て以下のように表現することができる。主体ＡとＢとでメッセージＭをDiffie-Hellman型暗号通信手順によって、暗号化して交換し、傍受者をＣで表わし、Diffie-Hellman型暗号通信手順に共通する「振る舞い」を「＋」で表現する。

【００７０】

暗号通信手順Ｐは、

Ｐ＝（Ｆ，｛Ｒｉ｝，Ｕ，｛Ｋｉ｝，Ｓ［1→3］）

と表現できる。ここで、ｉ＝１～３であり、各記号は、式１と同じ意味である。

【００７１】

関数記号の集合Ｆは、

Ｆ＝｛Ａ（０），Ｂ（０），Ｃ（０），Ｎ（０），Ｍ（０），ｋ（１），
＋（２），Ｅ（２），Ｄ（２）｝

となる。ここで、Ａ、Ｂは暗号情報を交換する主体、Ｃは傍受者であり、Ｎは任意の自然数、Ｍは暗号化された後に交換される秘密情報（例えば、初期に主体Ａのみが知っている情報で、暗号化されて主体Ｂに通信される情報）であり、ｋは鍵、Ｅは暗号処理、Ｄは復号処理を表す。「＋」は加算を表す記号ではなく、上記したようにDiffie-Hellman型暗号通信手順において共通する「振る舞い」を表す記号である。カッコ内の数字は各記号の引数、即ち決定に必要な「変数」の数である。従って、Ａ（０）、Ｂ（０）、Ｃ（０）、Ｎ（０）、Ｍ（０）は所定の記号又は数値であり、これらを決定するために変数は必要ない。ｋ（１）は、決定に１つの変数が必要であり、Ｅ、Ｄ、＋には各々２つの変数が必要である。例

えば、ｋ（１）は、主体Ａ、Ｂのいずれかを決めれば、ｋ（Ａ）又はｋ（Ｂ）として決定される。Ｅは、暗号処理の対象となる項と鍵とによって決定される。Ｄは、復号処理の対象となる項と鍵とによって決定される。＋は、ｍｏｄ関数の引数、即ち除数と被除数とによって決定される。

【００７２】

書換規則｛Ｒｉ｝は、

Ｒ１＝Ｒ２＝Ｒ３＝｛Ｄ（ｘ，Ｅ（ｘ，ｙ））→ｙ｝

である。これは、ｙを暗号処理した結果であるＥ（ｘ，ｙ）を復号処理すれば、ｙを得ることができることを表す。

【００７３】

公理Ｕは、

Ｕ＝｛ｘ＋ｙ＝ｙ＋ｘ，（ｘ＋ｙ）＋ｚ＝ｘ＋（ｙ＋ｚ）｝

である。ここでも、「＋」は加算ではなく、Diffie-Hellman型暗号通信手順において共通する「振る舞い」を表す。即ち、Diffie-Hellman型暗号通信手順において共通する「振る舞い」は、交換則、結合則を満たすことを表わしている。

【００７４】

ナレッジ｛Ｋｉ｝は、

Ｋ１＝｛Ａ，Ｂ，ｋ（Ａ）＋ｋ（Ｂ）＋Ｎ，ｋ（Ａ），Ｎ，Ｍ｝

Ｋ２＝｛Ａ，Ｂ，ｋ（Ａ）＋ｋ（Ｂ）＋Ｎ，ｋ（Ｂ），Ｎ，
Ｅ（ｋ（Ａ）＋ｋ（Ｂ）＋Ｎ，Ｍ）｝

Ｋ３＝｛Ａ，Ｂ，Ｃ，ｋ（Ａ）＋Ｎ，ｋ（Ｂ）＋Ｎ，ｋ（Ｃ），Ｎ，
Ｅ（ｋ（Ａ）＋ｋ（Ｂ）＋Ｎ，Ｍ）｝

である。

【００７５】

Ｋ１～Ｋ３は、それぞれ主体Ａ～Ｃのナレッジである。主体ＡとＢとの間で、鍵の情報としてＮ、ｋ（Ａ）＋Ｎ、ｋ（Ｂ）＋Ｎ、及び、暗号化されたメッセージＥ（ｋ（Ａ）＋ｋ（Ｂ）＋Ｎ，Ｍ）　が交換されることから、傍受者Ｃのナレッジ Ｋ３にはそれらの情報が含まれている。傍受者Ｃのナレッジ Ｋ３において、ｋ（Ａ）＋Ｎ、ｋ（Ｂ）＋Ｎ　と表記されているが、主体ＡとＢとの間の暗号通

信において、ｋ（Ａ）＋Ｎ、ｋ（Ｂ）＋Ｎ　はそれぞれ１つの情報として交換される。従って、傍受者Ｃはｋ（Ａ）＋Ｎを１つの情報として取得することは可能であるが、その構成、即ちｋ（Ａ）とＮとから生成されていることを直接知ることはできない。Ｅ（ｋ（Ａ）＋ｋ（Ｂ）＋Ｎ，Ｍ）に関しても同様に、傍受者ＣはＥ（ｋ（Ａ）＋ｋ（Ｂ）＋Ｎ，Ｍ）を１つの情報として取得できるだけで、その構成を直接知ることはできない。

【００７６】

主体Ａから傍受者Ｃに対する検査対象となる項の集合Ｓ［１→３］は、

Ｓ［１→３］＝｛Ｍ｝

である。

【００７７】

上記した代表的なDiffie-Hellman型暗号通信手順との対応関係は次の様になる。

・主体Ａが任意に選択した整数ｘは、ｋ（Ａ）に対応

・主体Ｂが任意に選択した整数ｙは、ｋ（Ｂ）に対応

・Ｘ（$X=g^x \bmod n$）は、ｋ（Ａ）＋Ｎに対応

・Ｙ（$Y=g^y \bmod n$）は、ｋ（Ｂ）＋Ｎに対応

・ｋ（$k=Y^x \bmod n$）は、ｋ（Ａ）＋（ｋ（Ｂ）＋Ｎ）に対応

・ｋ’（$k'=X^y \bmod n$）は、ｋ（Ｂ）＋（ｋ（Ａ）＋Ｎ）に対応

また、ｋ＝ｋ’の等価性から、ｋ（Ａ）＋（ｋ（Ｂ）＋Ｎ）＝ｋ（Ｂ）＋（ｋ（Ａ）＋Ｎ）が導かれる。ここで、ａ＋Ｎは、$g^a \bmod n$　を表すと仮定している。

【００７８】

以上で記述した、暗号通信手順Ｐ、関数記号の集合Ｆ、書換規則｛Ｒｉ｝、ナレッジ｛Ｋｉ｝、公理Ｕ、及び検査対象となる項の集合Ｓ［１→３］を、入力部５から入力することによって、図２に示したように、等式付ツリー・オートマトンが生成され、これを用いて計算が自動的に実行され、暗号通信手順の安全性が判断される。本実施例の場合には、上記したように、秘密情報は引数のない関数記号Ｍ（定数関数記号）として集合Ｆに含まれている。即ち、Ｍは項である。また

、検査対象はMだけからなる有限集合であるから、ステップ２６における判定の結果、ステップ２７の処理が実行される。即ち、暗号通信手順の安全性は、「Ａ３＊（傍受者Ｃが最終的に持ち得る知識の集合を受理する等式付ツリー・オートマトン）の中に、Ｍ（秘密情報）が含まれるか否か」という判断により行なわれる。

【００７９】

（実施例２）

本発明を、Shamirによって提案されたワンタイム・パッド（One-Time Pad）を用いた暗号通信手順に適用する場合について説明する。

【００８０】

ワンタイム・パッドの暗号通信手順は、交換則（Ｅ（ｋ（Ａ），Ｅ（ｋ（Ｂ），Ｍ））＝Ｅ（ｋ（Ｂ），Ｅ（ｋ（Ａ），Ｍ））を満たす暗号処理の下で、次の手順（１）～（４）で行われる。

（１）　主体Ａは、鍵ｋ（Ａ）を使用して秘密情報Ｍを暗号処理して得られたＥ（ｋ（Ａ），Ｍ）を、主体Ｂに送信する。

（２）　主体Ｂは、取得したＥ（ｋ（Ａ），Ｍ）を鍵ｋ（Ｂ）を使用して暗号処理して得られたＥ（ｋ（Ｂ），Ｅ（ｋ（Ａ），Ｍ））を、主体Ａに送信する。

（３）　主体Ａは、取得したＥ（ｋ（Ｂ），Ｅ（ｋ（Ａ），Ｍ））を鍵ｋ（Ａ）を使用して復号処理して得られたＤ（ｋ（Ａ），Ｅ（ｋ（Ｂ），Ｅ（ｋ（Ａ），Ｍ））），を、主体Ｂに送信する。

（４）　主体Ｂは、取得したＤ（ｋ（Ａ），Ｅ（ｋ（Ｂ），Ｅ（ｋ（Ａ），Ｍ））），を鍵ｋ（Ｂ）を使用して復号処理することによって秘密情報Ｍを取得することができる。即ち、交換則を考慮すれば、Ｄ（ｋ（Ｂ），Ｄ（ｋ（Ａ），Ｅ（ｋ（Ｂ），Ｅ（ｋ（Ａ），Ｍ））））＝Ｄ（ｋ（Ｂ），Ｄ（ｋ（Ａ），Ｅ（ｋ（Ａ），Ｅ（ｋ（Ｂ），Ｍ））））＝Ｄ（ｋ（Ｂ），Ｅ（ｋ（Ｂ），Ｍ））＝Ｍ　となる。

【００８１】

交換則を満たす暗号／復号処理として、暗号化しようとする平文に対して同じ長さの乱数ビット列を鍵として、これと平文との排他的論理和演算（以下、ＸＯ

Ｒと記す）が使用される。

【００８２】

この場合、暗号通信手順Ｐ、関数記号の集合Ｆ、公理Ｕ、及び検査対象となる項の集合Ｓ[１→３]は、「＋」がＸＯＲを表すとすれば、上記した実施例１の場合と同じ表記となる。

【００８３】

書換規則｛Ｒｉ｝は、

Ｅ（ｘ，Ｅ（ｙ，ｚ））　→　Ｅ（ｘ＋ｙ，ｚ）

Ｅ（ｉｄ，ｘ）　→　ｘ

ｉｄ＋ｉｄ　→　ｉｄ

（ｘ＋ｘ）＋ｙ　→　ｙ

と表わせる。ｉｄは、便宜上導入した関数である。

【００８４】

ナレッジ｛Ｋｉ｝は、

Ｋ１＝｛Ａ，Ｂ，ｋ（Ａ），ｉｄ，Ｍ，Ｅ（ｋ（Ｂ），Ｅ（ｋ（Ａ），Ｍ））｝

Ｋ２＝｛Ａ，Ｂ，ｋ（Ｂ），ｉｄ，Ｅ（ｋ（Ａ），Ｍ），Ｄ（ｋ（Ａ），Ｅ（ｋ（Ｂ），Ｅ（ｋ（Ａ），Ｍ））））｝

Ｋ３＝｛Ａ，Ｂ，Ｃ，ｋ（Ｃ），ｉｄ，Ｅ（ｋ（Ａ），Ｍ），Ｅ（ｋ（Ｂ），Ｅ（ｋ（Ａ），Ｍ）），Ｄ（ｋ（Ａ），Ｅ（ｋ（Ｂ），Ｅ（ｋ（Ａ），Ｍ））））｝

である。

【００８５】

以上で記述した、暗号通信手順Ｐ、関数記号の集合Ｆ、書換規則｛Ｒｉ｝、ナレッジ｛Ｋｉ｝、公理Ｕ、及び検査対象となる項の集合Ｓ［１→３］を、入力部５から入力することによって、図２に示したように、等式付ツリー・オートマトンが生成され、これを用いて計算が自動的に実行され、暗号通信手順の安全性が判断される。本実施例の場合にも、実施例１と同様に、秘密情報は引数のない関数記号Ｍとして集合Ｆに含まれていることから、ステップ２７の処理が実行される。

【００８６】

上記した実施例１、２において、無限の秘密情報を対象とする場合には、予め求められた秘密情報の集合を受理する等式付ツリー・オートマトンＳを入力することによって、ステップ２８～３０の処理によって安全性の検証が行われる。

【００８７】

以上において、検査対象となる項の集合Ｓ［ｊ→ｋ］が有限集合である場合には、ステップ２７の処理が行われるが、この場合にもＳ［ｊ→ｋ］を等式付ツリー・オートマトンとして記述して、予めステップ２０において入力しておくことによって、ステップ２８～３０の処理を行うようにすることができる。

【００８８】

また、以上において、検査対象となる項の集合が無限集合である場合には、ステップ２０において等式付ツリー・オートマトンとして記述されたＳ［ｊ→ｋ］を入力することとしたが、この場合にも検査対象となる項の集合Ｓ［ｊ→ｋ］を入力し、翻訳部８によってこれを受理する等式付ツリー・オートマトンを生成するようにすることができる。

【００８９】

また、以上においては、厳密な安全性の検証ができるように、図２のステップ２４の処理が収束しない場合、ステップ２６～３０の処理を行わないこととしたが、図５に示すように変更することも可能である。図５において、ステップ５０～５２、５６～６１の処理は、それぞれ図２のステップ２０～２２、２６～３１の処理と同じである。

【００９０】

ステップ５３において、ＣＰＵ１は、ステップ２３と同様にシミュレーション開始の命令コード等をシミュレーション部９に伝送し、シミュレーション部９はカウンタに初期値、例えば“０”をセットする。

【００９１】

ステップ５４において、シミュレーション部９は、ステップ２４での処理に関して説明したように、拡張されるナレッジＫｉを受理する等式付ツリー・オートマトンＡｉ＊（ｎ）を生成し、ステップ５３においてセットされたカウンタの初期値を１だけ変化、例えば１だけ増加させた後、ステップ５５に移行する。

【００９２】

ステップ５５において、シミュレーション部９は、カウンタの値が予め指定した値（例えば自然数ｎ０）であるか否かを判断する。指定値（ｎ０）であればステップ５６に移行し、指定値（ｎ０）でなければステップ５４に戻る。これによって、指定回数（ｎ０）だけステップ５４の処理を行なわせることができ、対応して拡張されたナレッジＫｉを受理する等式付ツリー・オートマトンＡｉ＊（ｎ０）が生成される。

【００９３】

図５においては、拡張されるナレッジＫｉを受理する等式付ツリー・オートマトンＡｉ＊（ｎ）の収束性を判断せずに、ステップ５４の処理を所定回数行った後、常にステップ５６以降の処理を行う。その結果、安全であると判断された場合には、その判断は必ずしも正しいものではなく、実際には安全でない可能性があるが、安全でないと判断された場合には、その判断は正しい。従って、図５に示した処理も、暗号通信手順の安全性の検証において有効な処理である。

【００９４】

以上において、暗号通信手順の安全性の検証に関して、実施の形態及び実施例を説明したが、その他のリアクティブ・システムに関しても同様に、安全性を検証することが可能である。例えば、原子炉や航空機などのシステムの制御手順に関する安全性を検証することが可能である。さらに、システムの規模によらず、設計の初期段階で安全性の検証をおこなうことができる。

【００９５】

この場合には、翻訳部８による初期の知識を受理する等式付ツリー・オートマトンを生成する処理（図２のステップ２２）は、対象のリアクティブ・システムを記述する等式付ツリー・オートマトンを生成する処理とすることができる。若しくは、予め生成された対象のリアクティブ・システムを記述する等式付ツリー・オートマトンをステップ２０で入力する場合には、ステップ２２における処理を省略することができる。

【００９６】

その他の処理は、上記した実施の形態と同様であり、例えば、有限の検査対象

となる状態、即ち危険な状態の集合Ｓ［ｊ→ｋ］が、ステップ２４において生成された等式付ツリー・オートマトンによって受理されるか否かを判断することによって、リアクティブ・システムの安全性の検証が可能となる。検査対象となる状態の集合Ｓ［ｊ→ｋ］が無限集合の場合には、等式付ツリーオートマトンとして記述されたＳ［ｊ→ｋ］を用いて、上記した実施の形態と同様に空の判定によって、リアクティブ・システムの安全性の検証が可能となる。

【００９７】

【発明の効果】

本発明に係るリアクティブ・システムの安全性検証装置によって、暗号通信手順に関して、通信の傍受者が取得可能な最大の知識の集合に対応する等式付ツリー・オートマトンを生成することができ、秘密情報が傍受者の知識の集合の中に含まれ得るか否か、即ち暗号通信手順の安全性を検証することが可能となる。特に、正則性の条件を満たさない暗号通信手順に関しても、近似ではなく、厳密に暗号通信手順の安全性を検証することが可能となる。

【００９８】

また、本発明に係るリアクティブ・システムの安全性検証装置において、傍受者の拡張される知識の集合に対応する等式付ツリー・オートマトンの収束性を判断しない場合でも、暗号通信手順が安全でないことを正確に判断することが可能となる。

【００９９】

本発明に係るリアクティブ・システムの安全性検証装置によって、一般のリアクティブ・システムに関して、取り得る状態の最大の集合を表す等式付ツリー・オートマトンを生成することができ、検査対象となる状態になり得るか否か、即ちリアクティブ・システムの安全性を検証することが可能となる。

【０１００】

大規模システムにおいては、設計の初期段階で安全性の検証をおこなうことができることから、検証によって発見された誤りによって設計変更を余儀なくされた場合でも、損失を小さく抑えることが可能となる。

【０１０１】

また、本発明に係るリアクティブ・システムの安全性検証装置において、一般のリアクティブ・システムに関して、拡張される状態の集合を表す等式付ツリー・オートマトンの収束性を判断しない場合でも、リアクティブ・システムが安全でないことを正確に判断することが可能となる。

【図面の簡単な説明】

【図１】　本発明の実施の形態に係るリアクティブ・システムの安全性検証装置の概略構成を示すブロック図である。

【図２】　本発明の実施の形態に係るリアクティブ・システムの安全性検証装置が行う処理を示すフローチャートである。

【図３】　本発明の実施の形態に係るリアクティブ・システムの安全性検証装置による、拡張された知識の集合を受理する等式付ツリー・オートマトンを計算する処理のアルゴリズムを示す説明図である。

【図４】　２つのオートマトンの結合処理を示す説明図である。

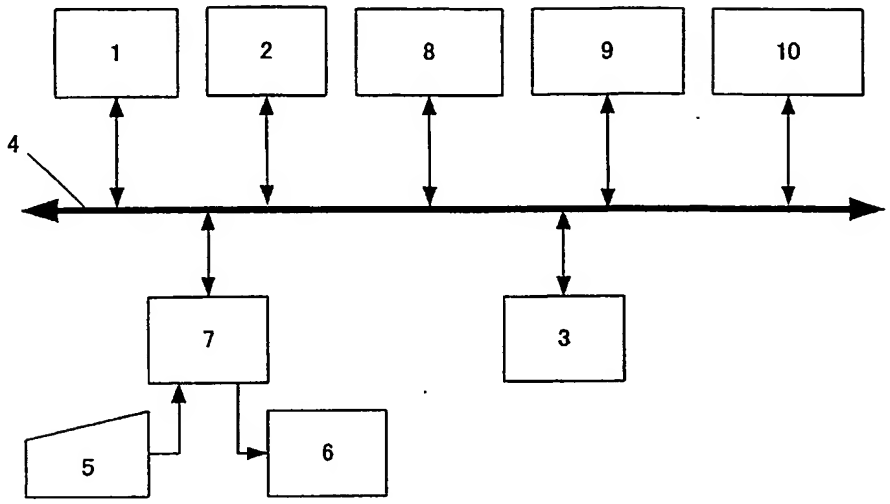
【図５】　本発明の実施の形態に係るリアクティブ・システムの安全性検証装置が行う処理において、拡張するナレッジを受理する等式付ツリー・オートマトンの収束性を判断しない場合の処理を示すフローチャートである。

【符号の説明】

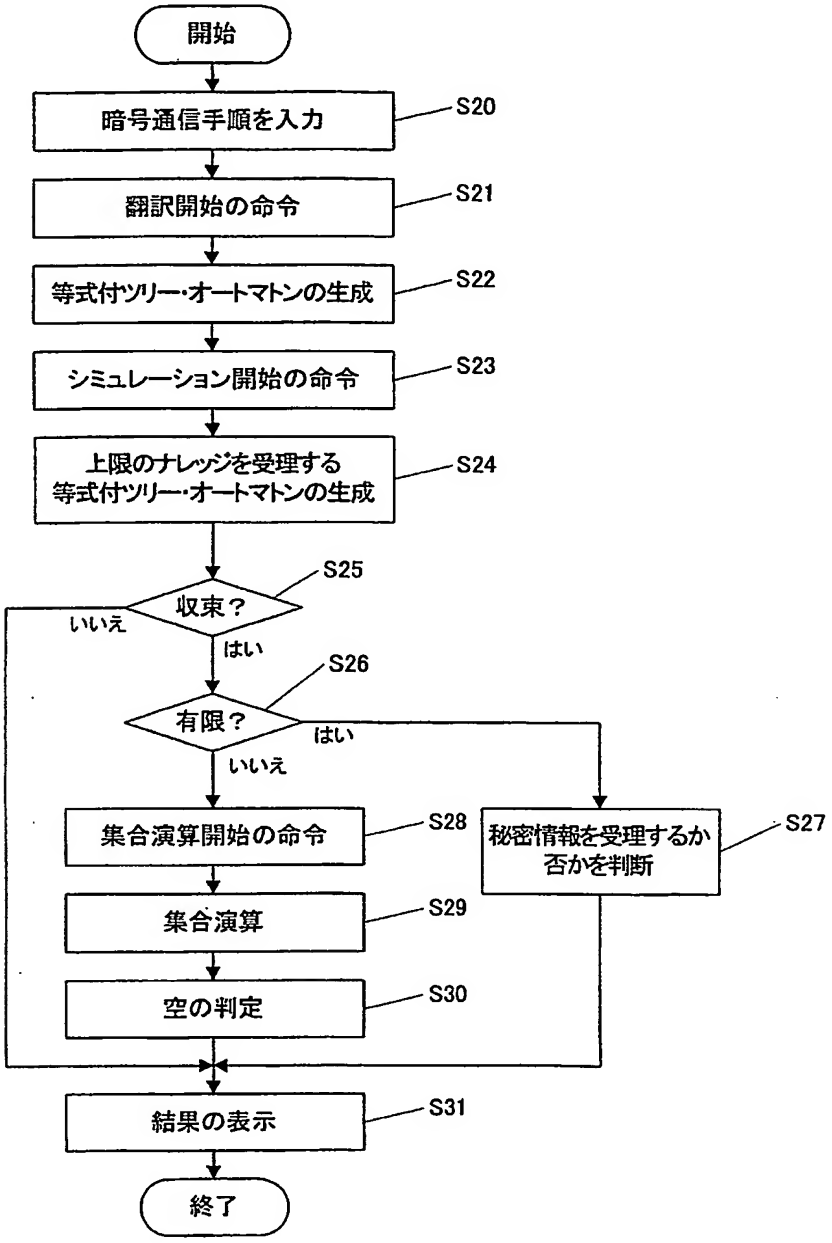
１　中央演算処理部（ＣＰＵ）
２　一時記憶部（メモリ）
３　記録部
４　内部バス
５　入力部
６　表示部
７　インタフェース部
８　翻訳部
９　シミュレーション部
１０　集合演算部

【書類名】　図面

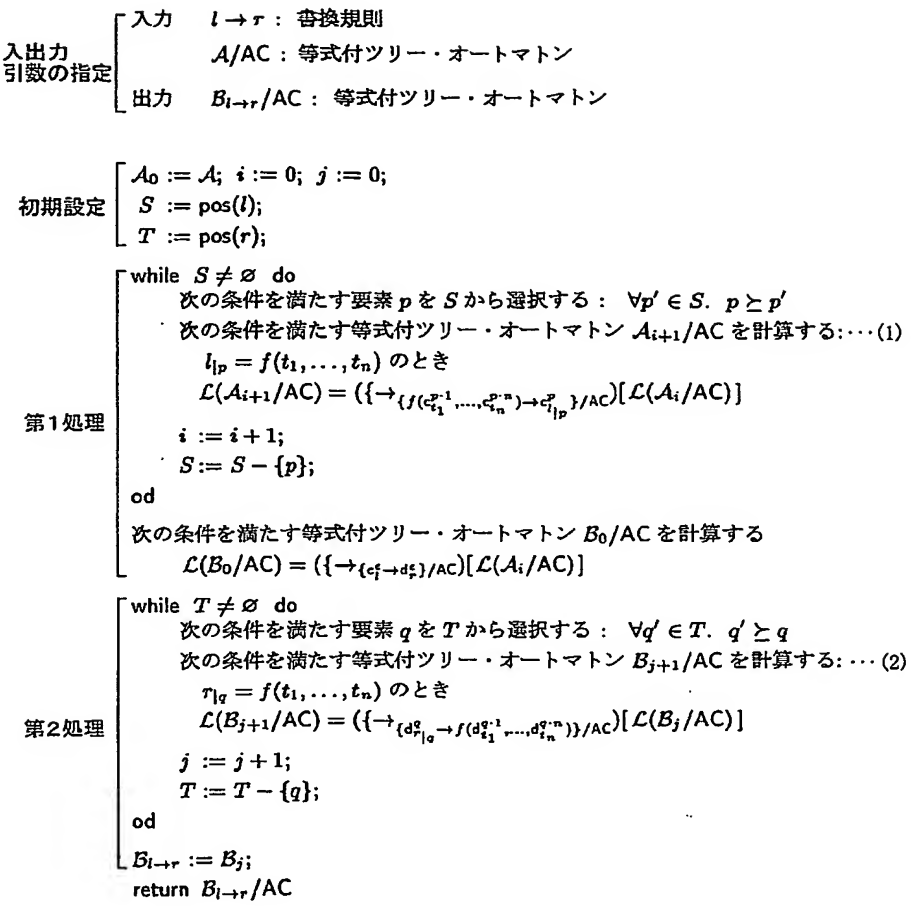
【図１】



【図２】


開始
暗号通信手順を入力
S20
翻訳開始の命令
S21
等式付ツリー・オートマトンの生成
S22
シミュレーション開始の命令
S23
上限のナレッジを受理する
等式付ツリー・オートマトンの生成
S24
S25
収束？
いいえ
はい
S26
有限？
はい
いいえ
集合演算開始の命令
S28
秘密情報を受理するか
否かを判断
S27
集合演算
S29
空の判定
S30
結果の表示
S31
終了

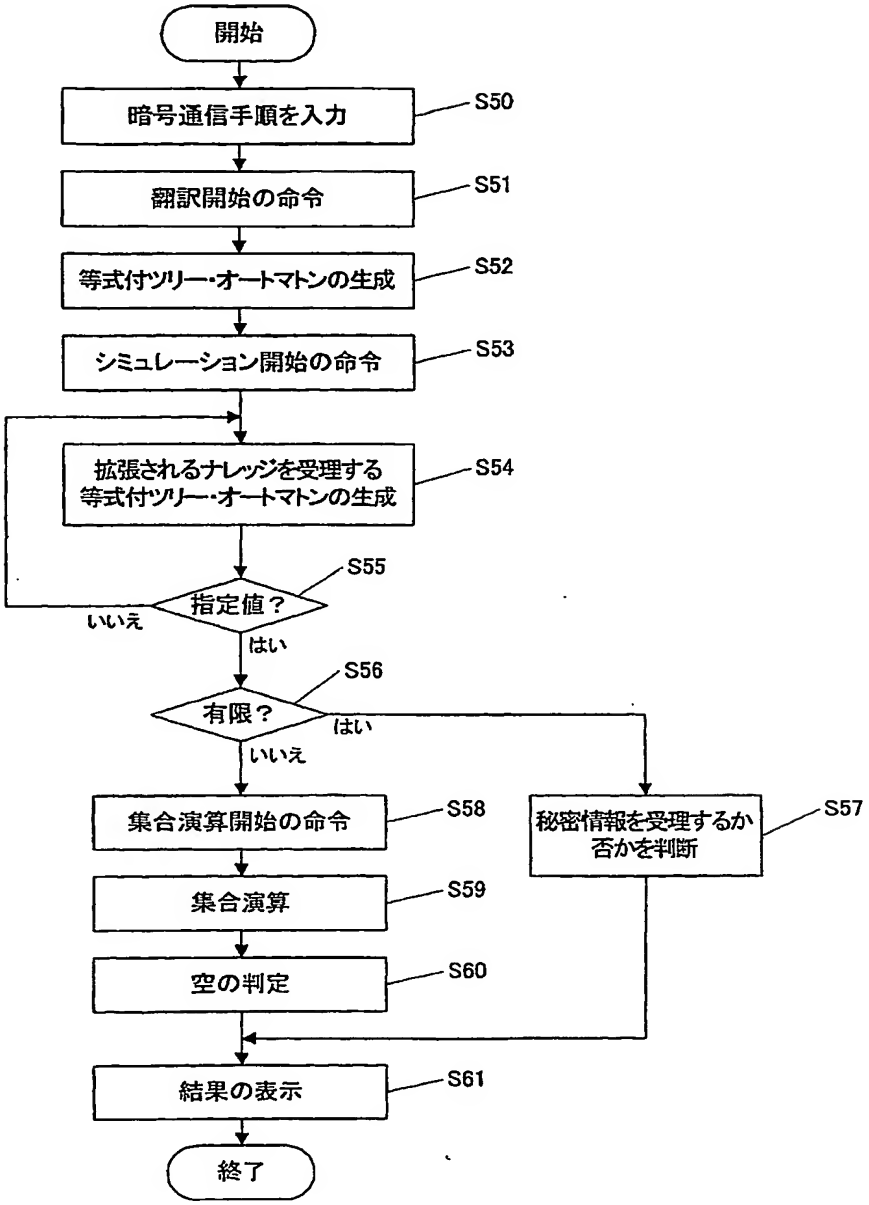
【図３】


入出力
引数の指定
入力　$l \to r$ ： 書換規則
$\mathcal{A}$/AC ： 等式付ツリー・オートマトン
出力　$\mathcal{B}_{l\to r}$/AC ： 等式付ツリー・オートマトン
初期設定
$\mathcal{A}_0 := \mathcal{A};\ i := 0;\ j := 0;$
$S := \mathsf{pos}(l);$
$T := \mathsf{pos}(r);$
第1処理
while $S \neq \varnothing$ do
次の条件を満たす要素 p を S から選択する： $\forall p' \in S.\ p \succeq p'$
次の条件を満たす等式付ツリー・オートマトン $\mathcal{A}_{i+1}$/AC を計算する:・・・(1)
$l_{|p} = f(t_1, \ldots, t_n)$ のとき
$\mathcal{L}(\mathcal{A}_{i+1}/\mathsf{AC}) = (\{\to_{\{f(c_{t_1}^{p\cdot 1},\ldots,c_{t_n}^{p\cdot n})\to c_{l|_p}^{p}\}/\mathsf{AC}}\}[\mathcal{L}(\mathcal{A}_i/\mathsf{AC})]$
$i := i+1;$
$S := S - \{p\};$
od
次の条件を満たす等式付ツリー・オートマトン $\mathcal{B}_0$/AC を計算する
$\mathcal{L}(\mathcal{B}_0/\mathsf{AC}) = (\{\to_{\{c_l^\varepsilon \to d_r^\varepsilon\}/\mathsf{AC}}\})[\mathcal{L}(\mathcal{A}_i/\mathsf{AC})]$
第2処理
while $T \neq \varnothing$ do
次の条件を満たす要素 q を T から選択する： $\forall q' \in T.\ q' \succeq q$
次の条件を満たす等式付ツリー・オートマトン $\mathcal{B}_{j+1}$/AC を計算する: ・・・(2)
$r_{|q} = f(t_1, \ldots, t_n)$ のとき
$\mathcal{L}(\mathcal{B}_{j+1}/\mathsf{AC}) = (\{\to_{\{d_{r_{|q}}^q \to f(d_{t_1}^{q\cdot 1},\ldots,d_{t_n}^{q\cdot n})\}/\mathsf{AC}}\})[\mathcal{L}(\mathcal{B}_j/\mathsf{AC})]$
$j := j+1;$
$T := T - \{q\};$
od
$\mathcal{B}_{l\to r} := \mathcal{B}_j;$
return $\mathcal{B}_{l\to r}$/AC

【図４】

集合	遷移規則	条件
$\mathcal{R}_\times$	$f((p_1,q_1),\ldots,(p_n,q_n)) \to (p,q)$	$\forall f \in \mathcal{F} \setminus \mathcal{G}$ $\forall f(p_1,\ldots,p_n) \to p \in \mathcal{R}_A$ $\forall f(q_1,\ldots,q_n) \to q \in \mathcal{R}_B$
$\mathcal{R}_{\overline{A}}$	$g((p_1,q_1),(p_2,q_2)) \to g((p,q_1),q_2)$ $g(p_1,(p_2,q_2)) \to (p,q_2)$	$\forall g \in \mathcal{G}$ $\forall q_1,q_2 \in \mathcal{Q}_B$ $\forall g(p_1,p_2) \to p \in \mathcal{R}_A$
	$g((p_1,q_1),(p_2,q_2)) \to g((r_1,q_1),(r_2,q_2))$ $g(p_1,(p_2,q_2)) \to g(r_1,(r_2,q_2))$	$\forall g(p_1,p_2) \to g(r_1,r_2) \in \mathcal{R}_A$
$\mathcal{R}_{\overline{B}}$	$g((p_1,q_1),(p_2,q_2)) \to g((p_1,q),p_2)$ $g(q_1,(p_2,q_2)) \to (p_2,q)$	$\forall g \in \mathcal{G}$ $\forall p_1,p_2 \in \mathcal{Q}_A$ $\forall g(q_1,q_2) \to q \in \mathcal{R}_B$
	$g((p_1,q_1),(p_2,q_2)) \to g((p_1,r_1),(p_2,r_2))$ $g(q_1,(p_2,q_2)) \to g(r_1,(p_2,r_2))$	$\forall g(q_1,q_2) \to g(r_1,r_2) \in \mathcal{R}_B$
$\mathcal{R}_\mathcal{G}$	$g((p,q_1),q_2) \to g(q_1,(p,q_2))$ $g((p_1,q),p_2) \to g(p_1,(p_2,q))$ $g(q,p) \to (p,q)$	$\forall g \in \mathcal{G}$ $\forall p_1,p_2,p \in \mathcal{Q}_A$ $\forall q_1,q_2,q \in \mathcal{Q}_B$

【図５】


開始
暗号通信手順を入力
S50
翻訳開始の命令
S51
等式付ツリー・オートマトンの生成
S52
シミュレーション開始の命令
S53
拡張されるナレッジを受理する
等式付ツリー・オートマトンの生成
S54
S55
指定値？
いいえ
はい
S56
有限？
はい
いいえ
集合演算開始の命令
S58
秘密情報を受理するか
否かを判断
S57
集合演算
S59
空の判定
S60
結果の表示
S61
終了

【書類名】　要約書

【要約】

【課題】　等式付ツリー・オートマトン理論の範疇にあるリアクティブ・システムの動作手順の安全性を検証すること。

【解決手段】　リアクティブ・システムの安全性検証装置において、公理の集合が交換則及び結合則のみを要素とし、前記公理の集合の下で、項の集合を受理する第１の等式付ツリー・オートマトンを生成する翻訳部８、前記第１の等式付ツリー・オートマトンを初期データとして、書換規則の集合及び前記公理の集合の下で、前記項の集合から派生する項からなる集合と前記項の集合とを受理する第２の等式付ツリー・オートマトンを生成するシミュレーション部９、及び前記第２の等式付ツリー・オートマトンと、検査対象となる項の集合を受理する第３の等式付ツリー・オートマトンとを結合して第４の等式付ツリー・オートマトンを生成し、該第４の等式付ツリー・オートマトンが受理する集合が空集合か否かを判断する集合演算部１０を備えている。

【選択図】　図１

特願２００２－２１１０２１

出　願　人　履　歴　情　報

識別番号　　［３０１０２１５３３］

１．変更年月日　　２００１年　４月　２日
［変更理由］　　新規登録
住　所　　東京都千代田区霞が関１－３－１
氏　名　　独立行政法人産業技術総合研究所